

Probabilistic bounded reachability for hybrid systems with continuous nondeterministic and probabilistic parameters

Fedor Shmarov and Paolo Zuliani

School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
 {f.shmarov, paolo.zuliani}@ncl.ac.uk

Abstract. We develop an algorithm for computing bounded reachability probability for hybrid systems, i.e., the probability that the system reaches an unsafe region within a finite number of discrete transitions. In particular, we focus on hybrid systems with continuous dynamics given by solutions of nonlinear ordinary differential equations (with possibly nondeterministic initial conditions and parameters), and probabilistic behaviour given by initial parameters distributed as continuous (with possibly infinite support) and discrete random variables. Our approach is to define an appropriate relaxation of the (undecidable) reachability problem, so that it can be solved by δ -complete decision procedures. In particular, for systems with continuous random parameters only, we develop a validated integration procedure which computes an arbitrarily small interval that is guaranteed to contain the reachability probability. In the more general case of systems with both nondeterministic and probabilistic parameters, our procedure computes a guaranteed enclosure for the range of reachability probabilities. We have applied our approach to a number of nonlinear hybrid models and validated the results by comparison with Monte Carlo simulation.

1 Introduction

Cyber-physical systems integrates digital computing (the *cyber* part) with a *physical* environment or device, in order to enhance or enable new capabilities of physical systems. Hybrid systems are mathematical models that combine continuous dynamics and discrete control, and enjoy widespread use for modelling cyber-physical systems. For example, Stateflow/Simulink¹ is the *de facto* standard tool for model-based design of embedded systems, and its semantics can be given in terms of hybrid systems (*e.g.*, [28]). Cyber-physical systems are used in many safety-critical applications, where a malfunctioning can result in threats to, or even loss of, human life. For example, modern aircraft are flown more efficiently by a computer, while anti-lock brakes and stability control contribute to safer cars. Again, electronic biomedical devices (*e.g.*, digital infusion pumps) offer superior flexibility and accuracy than traditional devices. Thus, verifying

¹ www.mathworks.com/simulink

safety of cyber-physical systems, and thereby of hybrid systems, is an extremely important problem.

The state space of a hybrid system consists of a discrete component and of a continuous component. The fundamental *reachability* problem is to decide whether a hybrid system reaches an *unsafe* region of its state space (a subset of states indicating incorrect behaviour of the system). Unfortunately, this problem is undecidable even for hybrid systems with constant differential dynamics [2]. For timed automata, *i.e.*, same constant differential dynamics across all the variables, the reachability problem is PSPACE-complete [3]. Also, it has been recently shown that bounded-time reachability of rectangular automata with non-negative rates is decidable [4]. However, hybrid systems arising from practical applications feature much richer dynamics, including non-linear functions over the reals, *e.g.*, trigonometric functions, for which even simple questions are in general undecidable [24]. Furthermore, for many practical applications it is necessary to augment hybrid systems with stochastic behaviour. Stochastic systems arise naturally when modelling phenomena which are intrinsically probabilistic, *e.g.*, soft errors in computing hardware. Also, stochastic systems can arise due to uncertainty in (deterministic) system components, its behaviours, and its environment. The reachability problem for stochastic hybrid systems asks what is the *probability* that the system reaches the unsafe region. (Note that for hybrid systems with both stochastic and non-deterministic behaviour the answer may be a range of probabilities.) In this work we focus on *bounded* reachability, *i.e.*, within a finite number of discrete transitions.

Since even standard reachability is undecidable, the problem must be modified if we want to solve it algorithmically. A possible solution is to relax it in a sound manner through the notions of δ -satisfiability and δ -complete decision procedures [11]. Such procedures sidestep undecidability by allowing a ‘tuneable’ precision in the answer provided. This is a necessary condition for decidability, and it motivates the notion of δ -satisfiability for logical formulae over the reals [11]. Using δ -satisfiability, in this paper we introduce and study the notion of probabilistic δ -reachability.

To summarise, in this paper:

- we formulate the bounded δ -reachability problem for hybrid systems with continuous/discrete probabilistic and nondeterministic initial parameters;
- we develop an algorithm that combines validated integration and δ -complete procedures into for computing a *numerically guaranteed* enclosure for the reachability probabilities. For models with continuous random (but no non-deterministic) parameters, such enclosure can be made *arbitrarily* small;
- we validate our algorithm against standard Monte Carlo probability estimation on a number of case studies.

Related Work. The SiSAT tool [9] solves probabilistic bounded reachability by returning answers guaranteed to be numerically accurate. However, SiSAT does not currently support continuous random parameters, while instead our tool does so (also with unbounded domains, *e.g.*, normal random variables). A very recent extension of SiSAT supports continuous nondeterminism, but the

technique is based on statistical model checking and therefore can only provide statistical guarantees [6], while we give numerical and formal guarantees. In [7] the authors present a technique for computing p-boxes using validated ODE integration. However, the technique is restricted to ODE systems and finite-support random parameters, while we handle hybrid systems and infinite-support random parameters. Moreover, it is not clear what guarantees are given for models containing only continuous and/or discrete random parameters: the size of the computed p-box might be quite large. In contrast, for continuous random parameters we can compute an arbitrarily small interval containing the exact reachability probability (see Proposition 11).

UPPAAL [18] is an extremely powerful model checker for timed automata, and it has been recently extended to support (dynamic) networks of stochastic timed automata via UPPAAL SMC [5]. However, UPPAAL SMC utilises a statistical model checking approach for reasoning about probabilities. PRISM [17] is a state-of-the-art model checker for a variety of discrete-state stochastic systems, but with respect to real-time systems it is limited to probabilistic timed automata. The tool FAUST² [27] utilises abstraction techniques to verify non-deterministic continuous-state Markov models, although currently for discrete-time models only. ProHVer computes an upper bound for the maximal reachability probability [30], and handles continuous random parameters via discrete over-approximation only [8]. We instead provide an enclosure (*both* upper and lower bounds) of the whole range of probabilities (for models with nondeterministic continuous parameters); in the case of continuous random parameters our enclosure can be arbitrarily tight (see Proposition 11). In [1] the authors introduce a technique for computing bounds on reachability probabilities for stochastic hybrid systems, using abstraction by discrete-time Markov chains. The technique is further extended to full LTL and nondeterminism [29]. In [23] the authors give model checking algorithms for PCTL formulae over continuous-time stochastic hybrid systems. However, in [1,29,23] continuous state space is handled through finite discretisation and approximated numerical solutions are provided for the experiments. We instead consider continuous time and space, and give full mathematical/numeric guarantees.

With respect to δ -satisfiability, in [21] the authors introduced and studied the complexity of a *relaxed* version of the verification problem, *i.e.*, verifying whether a given candidate is close to a problem solution. The (strong) verification problem is undecidable in general, so the authors relax it by introducing a “safety zone” in which either answer is deemed correct — this is δ -satisfiability.

Finally, in our work we use verified integration techniques (for an overview see, *e.g.*, [22] and references therein). Integration methods in the literature work with integrands in explicit form, *i.e.*, one must provide the actual mathematical expression for the integrand. Our approach is more general because: a) the integrand is given as a function of the numerical solution of possibly nonlinear ODEs; b) it considers hybrid dynamics. Our algorithm carries over the guarantees provided by δ -complete procedures for aspects a) and b) to the verified computation of a multi-dimensional integral over a possibly unbounded domain.

2 Probabilistic Bounded δ -Reachability

The following definition of hybrid system is a slight variant of the standard one.

Definition 1. *A hybrid system with probabilistic and nondeterministic initial parameters consists of the following components:*

- $Q = \{q_0, \dots, q_m\}$ a set of modes (discrete components of the system),
- $D = D_0 \times \dots \times D_p$ a domain of discrete random parameters, where each D_i is a finite set of reals,
- $R = [r_1, s_1] \times \dots \times [r_l, s_l] \subset \mathbb{R}^l$ a domain of continuous random parameters,
- $Z = [y_1, z_1] \times \dots \times [y_o, z_o] \subset \mathbb{R}^o$ a domain of nondeterministic parameters,
- $X = [u_1, v_1] \times \dots \times [u_n, v_n] \times [0, T] \subset \mathbb{R}^{n+1}$ a domain of continuous variables,
- $S = Q \times X$ is the hybrid state space of the system,
- $\Lambda = D \times R \times Z$ is the parameter space of the system,
- $U \subseteq S$ an unsafe region of the state space,

and predicates (or relations)

- $\text{flow}_q(\lambda, \mathbf{x}^0, \mathbf{x}^t)$ mapping the parameter $\lambda \in \Lambda$ and the continuous state \mathbf{x}^0 at time 0 to state \mathbf{x}^t at time point $t \in [0, T]$ in mode q
- $\text{init}_q(\mathbf{x}^0)$ indicating that $s = (q, \mathbf{x}^0)$ belongs to the set of initial states,
- $\text{jump}_{q \rightarrow q'}(\lambda, \mathbf{x}^t, \mathbf{x}^0)$ indicating that the system with parameter $\lambda \in \Lambda$ can make a transition from mode q , upon reaching the jump condition in continuous state \mathbf{x}^t at time $t \in [0, T]$, to mode q' and setting the continuous state to \mathbf{x}^0 ,
- $\text{unsafe}_q(\mathbf{x}^t)$ indicating that $s = (q, \mathbf{x}^t) \in U$

For all $q \in Q$ the sets defined by flow_q , init_q , jump_q , and unsafe_q are Borel; flow_q and jump_q are restricted to be functions of (λ, \mathbf{x}^0) and (λ, \mathbf{x}^t) , respectively.

The parameters in Λ are assigned in the initial mode and remain unchanged throughout the system's evolution. Also, the Borel assumption for the sets defined by the predicates is a theoretical requirement for well-definedness of probabilities, and in practice it is easily satisfied. The continuous dynamics of the system is defined in each flow, and it can either be presented as a system of Lipschitz-continuous ODEs or explicitly. In this paper we focus on hybrid systems for which in each mode only one jump is allowed to take place (of course the model may have multiple jumps, but only one jump should be enabled at any time). Given an initial value of the parameters, the semantics of a hybrid system can be informally thought as piece-wise continuous. (More details about the formal semantics can be found in [2].)

Bounded reachability asks whether the system reaches the unsafe region after $k \in \mathbb{N}$ discrete transitions.

Definition 2. [12] *The bounded k -step reachability property for hybrid systems with initial parameters is the bounded Σ_1 sentence $\exists \lambda \in \Lambda \ \psi(\lambda)$, where*

$$\begin{aligned} \psi(\lambda) = & \exists \mathbf{x}_{0,q_0}^0, \exists \mathbf{x}_{0,q_0}^t, \dots, \exists \mathbf{x}_{0,q_m}^0, \exists \mathbf{x}_{0,q_m}^t, \dots, \exists \mathbf{x}_{k,q_m}^0, \exists \mathbf{x}_{k,q_m}^t : \\ & (\bigvee_{q \in Q} (\text{init}_q(\mathbf{x}_{0,q}^0) \wedge \text{flow}_q(\lambda, \mathbf{x}_{0,q}^0, \mathbf{x}_{0,q}^t))) \\ & \wedge (\bigwedge_{i=0}^{k-1} (\bigvee_{q,q' \in Q} (\text{jump}_{q \rightarrow q'}(\lambda, \mathbf{x}_{i,q}^t, \mathbf{x}_{i+1,q'}^0))) \\ & \wedge (\text{flow}_{q'}(\lambda, \mathbf{x}_{i+1,q'}^0, \mathbf{x}_{i+1,q'}^t))) \wedge (\bigvee_{q \in Q} \text{unsafe}_q(\mathbf{x}_{k,q}^t))) \end{aligned} \quad (1)$$

Informally, the formula $\exists \lambda \in \Lambda \ \psi(\lambda)$ encodes the sentence “there exists a parameter vector for which starting from *init* and following *flow* and *jump*, the system reaches the unsafe region in k steps”. We obtain reachability *within* k steps by forming a disjunction of formula (1) for all values from 1 to k . The bounded reachability problem can be solved using a δ -complete decision procedure [11], which will *correctly* return one of the following answers:

- **unsat**: meaning that formula (1) is unsatisfiable (the system never reaches the bad region U);
- **δ -sat**: meaning that formula (1) is δ -satisfiable. In this case a witness, *i.e.*, an assignment for all the variables, is also returned.

With a δ -complete decision procedure, an **unsat** answer can always be trusted, while a **δ -sat** answer might in fact be a false alarm caused by the overapproximation. (In Appendix C we provide a short overview of δ -satisfiability.)

We now associate a probability measure to the random parameters, and we consider the following problem: what is the probability that a hybrid system with initial parameters reaches the unsafe region in k steps? Note that hybrid systems with both random and nondeterministic parameters will feature a range of reachability probabilities (although not necessarily a full interval).

Definition 3. *The probabilistic bounded k -step reachability problem for hybrid system with initial parameters is to compute an interval $[a, b]$ such that:*

$$\forall z_0 \in Z \quad \int_{B|_{z_0}} dP \in [a, b] \quad (2)$$

where

$$B = \{ \lambda \in \Lambda : \psi(\lambda) \} \quad (3)$$

and formula $\psi(\lambda)$ is per Definition 2; P is the probability measure associated with the random parameters; and $B|_{z_0}$ is the restriction of B to z_0 .

Informally, B is the set of the parameter values for which the system reaches the unsafe region in k steps.

Proposition 4. *The set B defined by (3) is Borel.*

(Proofs can be found in Appendix A.) The proposition entails that for any choice of the nondeterministic parameters, the probability that the system reaches the unsafe region is well-defined, and thereby Definition 3 is well-posed. When consider hybrid systems with continuous random parameters only, Definition 3 can be strengthened.

Definition 5. *Given any $\epsilon \in \mathbb{Q} \cap (0, 1]$, the probabilistic bounded k -step reachability problem for hybrid systems with random continuous initial parameters and single initial state is to compute an interval $[a, b]$ of length up to ϵ such that:*

$$\int_B dP \in [a, b] \quad (4)$$

where

$$B = \{\lambda \in \Lambda : \psi(\lambda)\} \quad (5)$$

and formula $\psi(\lambda)$ is per Definition 2; P is the probability measure associated with the random parameters.

Note that if only discrete random parameters are present it might not be possible to obtain an arbitrarily small enclosure. Also, in Definition 1 we require all continuous domains to be bounded: this is a necessary condition for δ -decidability of bounded reachability [11]. However, we later show that it is still possible to reason about random parameters with unbounded domains, *e.g.*, normally distributed. The key is that any probability density function can be approximated arbitrarily well by a truncation on a large (but finite) interval.

3 Validated Integration Procedure

We now present the first part of our δ -complete procedure for calculating the k -step reachability probability (4). The algorithm consists of a validated integration procedure and a decision procedure used for computing the set B of Definition 5. For clarity, we focus on one random continuous initial parameter.

Notation. For an interval $[r] = [\underline{r}, \bar{r}] \subset \mathbb{R}$ we denote the size of the interval by $width([r]) = \bar{r} - \underline{r}$ and by $mid([r]) = \frac{\bar{r} + \underline{r}}{2}$ the central point of the interval.

Our validated integration procedure employs the (1/3) Simpson rule:

$$K([I]) = \int_a^b f(x) dx = \frac{width([I])}{6} (f(\underline{I}) + 4f(mid([I])) + f(\bar{I})) - \frac{width([I])^5}{2880} f^{(4)}(\xi) \quad (6)$$

where $[I] = [a, b]$, $\xi \in [I]$ and $f^{(4)}$ is the fourth derivative of an integrable function f . For our applications the integrands are probability density functions, which satisfy the required integrability and differentiability conditions. Our aim is to compute an interval of arbitrary size $\epsilon \in (0, 1] \cap \mathbb{Q}$ that contains K .

Definition 6. An interval extension of function $f : X \rightarrow Y$ is an operator $[\cdot]$ such that:

$$\forall x \in [r] \subseteq X : f(x) \in [f]([r]) \subseteq Y$$

By applying interval arithmetics, one computes interval extensions of f and $f^{(4)}$. (Interval extensions can be computed using interval arithmetics libraries, *e.g.*, FILIB++ [19].) The interval version of Simpson's rule can be obtained simply by replacing in (6) the occurrences of f and $f^{(4)}$ with their interval extension $[f]$, and by replacing ξ with the entire interval I [10]:

$$K \in [K]([I]) = \frac{\text{width}([I])}{6} ([f](I) + 4[f](\text{mid}([I])) + [f](\bar{I})) - \frac{\text{width}([I])^5}{2880} [f]^{(4)}([I]).$$

Furthermore, by the definition of integral:

$$K \in \Sigma_{i=1}^n [K]([x]_i) \quad (7)$$

where the collection of $[x]_i$'s is a partition of $[a, b]$. Note that we require a partition in a measure-theoretic sense, *i.e.*, intersections have (Lebesgue) measure 0, since these have no effect on integration.

In order to guarantee ϵ -completeness of the integration it is sufficient to partition $[a, b]$ into n intervals $[x]_i$ such that for each $[x]_i$ we have $\text{width}([I]([x]_i)) < \epsilon \frac{\text{width}([x]_i)}{b-a}$. Then, the exact value K of the integral will belong to an interval (7) of width smaller than ϵ . Pseudo-code for the procedure computing integral (6) up to an arbitrary $\epsilon \in (0, 1] \cap \mathbb{Q}$ is given in Algorithm 1. For our purposes we will only make use of the interval partition T , which will enable us to compute the reachability probability, *i.e.*, integral (4), with precision ϵ .

Proposition 7. If $f \in P_{C^5[a,b]}$, then the complexity of Algorithm 1 is NP.

4 Computing δ -Reachability Probability

4.1 Computing indicator functions

From Algorithm 1 we obtain a partition of the domain of the random parameters which will guarantee the computation of integral (4) with the desired accuracy. In general, given $z_0 \in Z$, the reachability probability is computed by integrating the probability measure of the random parameters over the restriction $B|_{z_0}$. We need to compute the following integral

$$\int_{B|_{z_0}} dP(r) \quad \left(= \int_{D \times R} I_U(r, z_0) dP(r) \right)$$

where B is the set (3), $z_0 \in Z$, and I_U is the indicator function

$$I_U(r, z_0) = \begin{cases} 1 & \text{if the system with parameter } (r, z_0) \text{ reaches } U \text{ in } k \text{ steps} \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 1: Validated Integration Procedure

```

input: function  $f$ , interval  $[a, b]$ ,  $\epsilon \in \mathbb{Q}^+$ ;
output:  $[I]$ , partition  $T$  of  $[a, b]$  such that  $\int_a^b f \in [I]$  and  $\text{width}([I]) \leq \epsilon$ ;
 $[I] = [0.0, 0.0]$ ;
 $T, B = \emptyset$ ;
// put initial partition on a stack
 $B.\text{push}(\{[a, b], [K]([a, b])\})$ ;
while  $\text{size}(B) > 0$  do
   $\{[x], [y]\} = B.\text{pop}()$ ;
  if  $\text{width}([y]) > \epsilon \frac{\text{width}([x])}{b-a}$  then
    // split the interval in two
     $B.\text{push}(\{[x, \text{mid}([x])], [K]([x, \text{mid}([x])])\})$ ;
     $B.\text{push}(\{[\text{mid}([x]), \bar{x}], [K]([\text{mid}([x]), \bar{x}])\})$ ;
  else
    // add sub-integral to the partial sum; save interval
     $[I] = [I] + [K]([x])$ ;
     $T.\text{push}([x])$ ;
return  $T, [I]$ ;

```

We now show how to compute I_U or, equivalently, set B . Let $[\rho] \subseteq A$ be a box and ϕ be a formula of the form:

$$\phi([\rho]) = \exists \lambda \in [\rho] \psi(\lambda) \quad (8)$$

If the formula is true then $[\rho]$ contains a value for the initial parameters for which the system reaches the unsafe region U . Taking the complement of the unsafe region $U^C = S/U$ (S is the state space of the system) and defining a predicate $\text{unsafe}_q^C(\mathbf{x}^t) \equiv ((q, \mathbf{x}_q^t) \in U^C)$ we want to ensure that the system never reaches the unsafe region *within* the k -th step with an initial parameter from $[\rho]$. In order to conclude that it is sufficient to evaluate the formula:

$$\begin{aligned}
\phi^C([\rho]) = & \exists \lambda \in [\rho], \exists \mathbf{x}_{0,q_0}^0, \exists \mathbf{x}_{0,q_0}^t, \exists t_{0,q_0}, \dots, \exists \mathbf{x}_{0,q_m}^0, \\
& \exists \mathbf{x}_{0,q_m}^t, \exists t_{0,q_m}, \dots, \exists \mathbf{x}_{k,q_m}^0, \exists \mathbf{x}_{k,q_m}^t, \exists t_{k,q_m}, \forall t'_{k,q_m} \in [0, t_{k,q_m}] : \\
& \left(\bigvee_{q \in Q} (\text{init}_q(\mathbf{x}_{0,q}^0) \wedge \text{flow}_q(\lambda, \mathbf{x}_{0,q}^0, \mathbf{x}_{0,q}^t)) \right) \wedge \\
& \left(\bigwedge_{i=0}^{k-1} \left(\bigvee_{q, q' \in Q} (\text{jump}_{q \rightarrow q'}(\lambda, \mathbf{x}_{i,q}^t, \mathbf{x}_{i+1,q'}^0) \wedge \right. \right. \\
& \quad \left. \left. (\text{flow}_{q'}(\lambda, \mathbf{x}_{i+1,q'}^0, \mathbf{x}_{i+1,q'}^t)) \right) \right) \wedge \\
& \left(\bigvee_{q \in Q} (\text{unsafe}_q^C(\mathbf{x}_{k,q}^t) \wedge (\text{jump}_{q \rightarrow q'}(\lambda, \mathbf{x}_{k,q}^t, \mathbf{x}_{k+1,q'}^0) \vee (t_{k,q_m} \geq T))) \right)
\end{aligned} \quad (9)$$

Note that ϕ^C is not the logical negation of ϕ — it is in fact an $\exists\forall$ -quantified formula. The last term of ϕ^C ensures that the system either does not reach the

unsafe region on the k -th step before it can make a transition to the successor mode or it reaches the time bound before reaching the unsafe region. This should not be confused with reaching the time bound in any of the preceding modes as it means that the system fails to reach the k -th step and should be, therefore, unsatisfiable. If the formula evaluates to true then the system does not reach the unsafe region on the k -th step. Then, set B can be defined as a finite collection $\{[\rho]_i : \phi([\rho]_i) \wedge (\neg\phi^C([\rho]_i))\}$. To build such a collection, we iteratively evaluate ϕ and ϕ^C with a δ -complete procedure (e.g., dReal [13]). Given a box $[\rho]$, there are four possible outcomes:

- $\phi([\rho])$ is **unsat**. Hence, there are *for sure* no values in $[\rho]$ such that the system reaches the unsafe region, so $[\rho]$ is not in B .
- $\phi([\rho])$ is **δ -sat**. Then, there is a value in $[\rho]$ such that the system reaches U or U^δ (δ -weakening of set U).
- $\phi^C([\rho])$ is **unsat**. Therefore, there is *for sure* no value in $[\rho]$ such that for all time points on the k -th step the system stays in U^C . In other words, for all the values in $[\rho]$ the system reaches U , so $[\rho]$ is fully contained in B .
- $\phi^C([\rho])$ is **δ -sat**. Then there is a value in $[\rho]$ such that the system stays within U^C or U^{C^δ} . In combination with outcome **δ -sat** for $\phi([\rho])$ it signals that $[\rho]$ is a *mixed* interval (it contains values from both B and B^C).

Therefore, **unsat** answers enable us to decide whether $[\rho]$ is a subset of or disjoint from set B . If **δ -sat** is returned for both formulae, then we are either dealing with a false alarm (an unsatisfiable formula is verified as **δ -sat** because of the overapproximation) or a mixed interval.

4.2 Main algorithm

The overapproximation (controlled by δ) introduced by δ -complete procedures can cause false alarms. We thus begin by addressing the choice of δ . Obviously, it is impossible to decide *correctly* (i.e., obtaining **unsat** for one of ϕ and ϕ^C) on each interval if a fixed δ (even a very small one) is used for evaluate all formulae.

Lemma 8. *Let ϕ be an arbitrary bounded Σ_1 formula and ϕ^δ its weakening. Then the following holds:*

$$\forall \delta, \delta' \in \mathbb{Q}^+, 0 \leq \delta' < \delta : \neg\phi^\delta \rightarrow \neg\phi^{\delta'} \rightarrow \neg\phi$$

(See Appendix C for an overview of δ -weakening.) Lemma 8 means that unsatisfiability of a weakened formula implies unsatisfiability of its strengthening and of the initial formula. We next show that when an interval is uncertain, by applying Lemma 8 we can obtain δ and a subinterval for which a δ -complete decision procedure can give a correct answer.

Proposition 9. *Let ϕ and ϕ^C as per (8) and (9), and $[u, v]$ an interval. Then:*

$$\begin{aligned} \exists \delta \in \mathbb{Q}^+ : (\phi([u, v]) - \delta\text{-sat}) \wedge (\phi^C([u, v]) - \delta\text{-sat}) \Rightarrow \\ \exists [u', v'] \subseteq [u, v] : (\phi([u', v']) - \text{unsat}) \oplus (\phi^C([u', v']) - \text{unsat}) \end{aligned}$$

where \oplus denotes exclusive or.

We now present the full algorithm for computing bounded reachability probability. We begin by addressing random initial parameters with (un)bounded support. Given $\epsilon \in (0, 1] \cap \mathbb{Q}$, it is always possible to find a *bounded* region of the random variable support with area larger than $1 - \epsilon$. In fact, such a problem can be stated as a δ -satisfiability question and thus solved by a δ -complete procedure. Therefore, the verified integration procedure presented in Section 3 can be applied to a random variable with unbounded domain. If we introduce multiple independent random parameters we can still use the same verified integration procedure provided that each random variable is integrated with a higher accuracy, as the next proposition shows.

Proposition 10. *Given a hybrid system with l independent continuous random parameters, to compute with precision $\epsilon_{prod} \in (0, 1] \cap \mathbb{Q}$ the reachability probability it is sufficient that each random variable is integrated with precision ϵ satisfying:*

$$\epsilon_{prod} \geq \sum_{i=1}^l \binom{l}{i} \epsilon^i \quad (10)$$

where $\binom{l}{i}$ is the binomial coefficient.

Suppose now a hybrid system has (continuous) nondeterministic parameters. Then the probability that the system reaches the unsafe region becomes a function of the nondeterministic parameters. In particular, the indicator function $I_U(r, z)$ can be equal to 0 and 1 for the same values of the continuous *random* parameters, *i.e.*, there may exist r_0 and $z_0 \neq z_1$ such that $I_U(r_0, z_0) = 0$ and $I_U(r_0, z_1) = 1$. Therefore, it is in general impossible to provide any guarantees on the length of probability interval, and we need to compute an enclosure for all probabilities. We will use the following symbolic notation for hybrid systems:

- **HA** (*Hybrid Automaton*) - a hybrid system without initial random parameters (only deterministic and nondeterministic).
- **PHA** (*Probabilistic Hybrid Automaton*) - a hybrid system with random and deterministic continuous initial parameters (no nondeterminism).
- **NPHA** (*Nondeterministic Probabilistic Hybrid Automaton*) - a hybrid system with random, deterministic and nondeterministic continuous initial parameters.

We first state the algorithm for **NPHAs** with no discrete probability.

Proposition 11. *Given $\epsilon \in (0, 1] \cap \mathbb{Q}$, $k \in \mathbb{N}$, and an **NPHA** without discrete random parameters, there exists an algorithm for computing an interval containing the set of k -step reachability probabilities. If the system has no nondeterministic parameters, the algorithm returns an interval of size not larger than ϵ containing the k -step reachability probability (4).*

The pseudo-code of the algorithm is presented in Algorithm 2. Informally, the algorithm starts by getting an interval partition from the validated integration procedure (Algorithm 1) for each random variable; also, a candidate probability interval is initialised to $[0, 1]$. Then, it evaluates the formulae ϕ and ϕ^C on

the current partition, which will be refined whenever both ϕ and ϕ^C are δ -**sat**. Instead, an **unsat** answer is used to refine the probability interval. The termination condition depends on the model type. If there are no nondeterministic parameters, then the algorithm will terminate when the width of the probability interval satisfies the desired size ϵ . Otherwise, the algorithm terminates when the maximum length of the boxes in the partition is smaller than ϵ . (Given a box we can split it into 2^n boxes of (pairwise) equal size in such a way that each interval in the box is reduced. However, any division strategy can be applied as long as the size of each interval forming the box is reduced.)

Theorem 12. *Given $\epsilon \in (0, 1] \cap \mathbb{Q}$, $k \in \mathbb{N}$, and a full **NPHA**, there exists an algorithm for computing an interval containing the set of k -step reachability probabilities. If the system has no nondeterministic parameters, the algorithm returns an interval of size not larger than ϵ containing the k -step reachability probability (4).*

Algorithm 3 drives the whole verification loop, while also handling discrete random parameters (with essentially the same technique as before). Notice that when the model has continuous parameters, Algorithm 2 is utilised.

Theorem 13. *The complexity of Algorithm 2 is $NP^{(\Sigma_2^P)^C}$, where $P \subseteq C \subseteq PSPACE$ is the complexity of the terms in the description of the hybrid system. With Lipschitz-continuous ODEs terms the complexity is $PSPACE$ -complete.*

5 Experiments

We have implemented our algorithms in **ProbReach**; its source code and the models studied are on <https://github.com/dreal/probreach>. (The tool implementation is explained in [25].) The results below can be also accessed on <https://homepages.ncl.ac.uk/f.shmarov/probreach>. All experiments were carried out on a multi-core Intel Xeon E5-2690 2.90GHz system running Linux Ubuntu 14.04LTS. The algorithms were also parallelised, and the results below feature the **ProbReach** CPU time of the parallel version on 24 cores.

We have applied **ProbReach** to four hybrid models: a 2D-moving bouncing ball, human starvation, prostate cancer therapy, and car collision scenario. The models feature a variety of highly nontrivial dynamics. For example, the ODEs for the prostate cancer therapy model [20] include exponential terms:

$$\begin{aligned}\frac{dx}{dt} &= \left(\frac{\alpha_x}{1 + e^{(k_1 - z)k_2}} - \frac{\beta_x}{1 + e^{(z - k_3)k_4}} - m_1 \left(1 - \frac{z}{z_0} \right) - c_1 \right) x + c_2 \\ \frac{dy}{dt} &= m_1 \left(1 - \frac{z}{z_0} \right) x + \left(\alpha_y \left(1 - d_0 \frac{z}{z_0} \right) - \beta_y \right) y \\ \frac{dz}{dt} &= -z\gamma - c_3\end{aligned}$$

More details on the models used in the experiments and the actual **ProbReach** model file for the prostate cancer therapy can be found in Appendix B.

Algorithm 2: Probabilistic δ -reachability PHA and NPHA

```
input: continuous random parameters  $\bar{r} = \{r_1, \dots, r_l\}$  with their probability
densities  $\bar{f}(r)$ ,  $\epsilon_{prod} \in (0, 1] \cap \mathbb{Q}$ , hybrid system description  $\phi$ ;
output: interval  $[I]$  enclosing the reachability probabilities
// obtain  $\epsilon$  from (10) using  $\epsilon_{prod}$ 
 $\epsilon_{inf} = t\epsilon$ ;
 $\epsilon_{prob} = (1 - t)\epsilon$ ;
// obtain bounds from (21) for each continuous random parameter
 $[\bar{a}, \bar{b}] = \bigcup_{i=1}^l (\mathbf{bounds}(f(r_i), \epsilon_{inf}))$ ;
 $[\bar{r}] = \bigcup_{i=1}^l (\mathbf{Algorithm1}(f(r_i), [a_i, b_i], \epsilon_{prob}))$ ;
// get Cartesian product of intervals from obtained partitions
 $B.\text{push}([r_1] \times \dots \times [r_l])$ ;
// set initial probability intervals for upper/lower approximation
 $[P_{lower}] = [0.0, 0.0]$ ;
 $[P_{upper}] = [1.0, 1.0]$ ;
// consider unbounded segments
 $[P_{upper}] = [P_{upper}] + 1 - \prod_{i=1}^l \int_{a_i}^{b_i} f_i(x) dx$ ;
while true do
    // stack containing extra divisions of the boxes
     $D = \emptyset$ ;
    while  $\text{size}(B) > 0$  do
         $\mathbf{box} = B.\text{pop}()$ ;
        //  $\delta$ -complete procedure evaluates formula
        if  $\phi(\mathbf{box}) = \delta\text{-sat}$  then
            //  $\delta$ -complete procedure evaluates formula
            if  $\phi^C(\mathbf{box}) = \delta\text{-sat}$  then
                // split initial box into  $2^l$  boxes of equal size
                 $D.\text{push}(\text{branch}(\mathbf{box}))$ ;
            else
                // increase lower approximation
                 $[P_{lower}] = [P_{lower}] + [S](\mathbf{box})$ ;
            else
                // decrease upper approximation
                 $[P_{upper}] = [P_{upper}] - [S](\mathbf{box})$ ;
         $B = D$ ;
    if  $\text{MODEL\_TYPE}(\phi) = \mathbf{PHA}$  then
        // termination condition when nondeterminism is absent
        if  $\overline{[P_{upper}]} - [P_{lower}] \leq \epsilon_{prob}$  then
            return  $[[P_{lower}], [P_{upper}]]$ ;
    if  $\text{MODEL\_TYPE}(\phi) = \mathbf{NPHA}$  then
        if  $\max_{\mathbf{box} \in B} (|\mathbf{box}|) \leq \epsilon$  then
            return  $[[P_{lower}], [P_{upper}]]$ ;
```

Algorithm 3: Main ProbReach algorithm

```
input: continuous random parameters  $\bar{r} = \{r_1, \dots, r_l\}$  with probability densities  $\bar{f}(r)$ , discrete random parameters  $\{D_1, \dots, D_p\}$  with probability distributions  $\bar{p}(\cdot)$ ,  $\epsilon_{prod} \in (0, 1] \cap \mathbb{Q}$ , hybrid system description  $\phi$ ;  
output: interval  $[I]$  enclosing reachability probability  
// obtain Cartesian product of discrete random parameters  
DD.push( $D_1 \times \dots \times D_p$ );  
// set initial probability intervals for upper/lower approximation  
 $P = \emptyset$ ;  
while size(DD) > 0 do  
  dd = DD.pop();  
   $m_{dd} = \prod_{d \in dd} p(d)$ ;  
  // syntactically replace discrete parameters and get new model  
  // if no other parameters are present,  $\phi_{dd}$  becomes an HA  
   $\phi_{dd} = \phi[(D_1, \dots, D_p) \setminus dd]$ ;  
  if MODEL_TYPE( $\phi_{dd}$ ) = HA then  
    //  $\delta$ -complete procedure evaluates formula  
    if  $\phi_{dd} = \delta\text{-sat}$  then  
      //  $\delta$ -complete procedure evaluates formula  
      if  $\phi_{dd}^C = \delta\text{-sat}$  then  
        // could not correctly decide for  $\phi_{dd}$   
        P.push( $m_{dd} \cdot [0.0, 1.0]$ );  
      else  
        // formula  $\phi_{dd}$  is sat  
        P.push( $m_{dd} \cdot [1.0, 1.0]$ );  
      else  
        // formula  $\phi_{dd}$  is unsat  
        P.push( $m_{dd} \cdot [0.0, 0.0]$ );  
    if (MODEL_TYPE( $\phi_{dd}$ ) = PHA  $\vee$  NPHA) then  
       $[I] = \text{Algorithm 2}(\bar{r}, \epsilon_{prod}, \phi_{dd})$ ;  
      // add obtained probability interval to the stack  
      P.push( $m_{dd} \cdot [I]$ );  
  // obtain sum of all probability intervals on the stack  
   $[[P_{lower}], [P_{upper}]] = \sum_{[I] \in P} [I]$ ;  
return  $[[P_{lower}], [P_{upper}]]$ ;
```

All experiments were validated using Monte Carlo probability estimation in MATLAB (the reported CPU times are for one core). In particular, we calculated confidence intervals using the Chernoff-Hoeffding bound [14]. All results are given in the tables below, where the top half of each table contains the results obtained using our approach (**ProbReach**), while the bottom half reports the Monte Carlo results. For space reasons, the results of the bouncing ball are presented in the Appendix B. Monte Carlo simulation of continuous nondeterminism (for **NPHA** models) was implemented by first uniformly discretising the domain of the nondeterministic parameters. Then, for each (discretised) value of the parameters we built a confidence interval using the Chernoff-Hoeffding bound. Finally, the Monte Carlo interval reported in the tables below is the union of all such confidence intervals. Note that Monte Carlo intervals for **NPHAs** will be in general larger than 2ζ , and they will not be proper confidence intervals because of the nondeterministic parameters. From the results we can see that our technique performs well even on highly nonlinear ODEs models such as the prostate cancer treatment model, despite having unavoidably high complexity (see Theorem 13). All Monte Carlo intervals cover the enclosures computed by **ProbReach**, thus confirming the correctness of our algorithms and their implementation.

Table legend: k = number of discrete transitions; ϵ = desired size of probability interval (**PHAs** only; lower box size limit for **NPHA**); $length$ = length of probability interval returned by **ProbReach**; ζ, c = half-interval width and coverage probability for Chernoff bound; N = sample size from Chernoff bound; CPU = CPU time (sec).

Table 1. Starvation model; see legend in Section 5.

Method	Model type	k	ϵ	$length$	Probability interval	CPU
Prob	NPHA	0	10^{-3}	$4.245 \cdot 10^{-3}$	[0.9219413, 0.92618671]	1,152
Reach	PHA	0	10^{-3}	$6.795 \cdot 10^{-4}$	[0.92455817, 0.92523768]	23

Method	Model type	k	ζ	c	Monte Carlo interval	CPU	N
Monte	NPHA	0	$5 \cdot 10^{-3}$	0.99	[0.9179, 0.9311]	12,433	92,104
Carlo	PHA	0	$5 \cdot 10^{-3}$	0.99	[0.9193355, 0.9293355]	2,868	92,104

Table 2. Prostate cancer therapy model; see legend in Section 5.

Method	Model type	k	ϵ	$length$	Probability interval	CPU
Prob	PHA	1	10^{-3}	$6.022 \cdot 10^{-4}$	[0.47380981, 0.47441201]	737
Reach	NPHA	1	10^{-4}	$1.763 \cdot 10^{-3}$	[0.4725522, 0.47431526]	89,925

Method	Model type	k	ζ	c	Monte Carlo interval	CPU	N
Monte	PHA	1	$1 \cdot 10^{-2}$	0.99	[0.4648111, 0.4848111]	5,700	23,026
Carlo	NPHA	1	$1 \cdot 10^{-2}$	0.99	[0.4583, 0.4890]	12,309	23,026

6 Conclusions and Future Work

We have given a formal definition of the bounded probabilistic δ -reachability problem for hybrid systems with continuous random and nondeterministic initial

Table 3. Car collision model; see legend in Section 5.

Method	Model type	k	ϵ	$length$	Probability interval	CPU
ProbReach	PHA	4	10^{-3}	$8.369 \cdot 10^{-4}$	$[0.5063922, 0.5072291]$	1,869

Method	Model type	k	ζ	c	Monte Carlo interval	CPU	N
Monte Carlo	PHA	4	$5 \cdot 10^{-3}$	0.99	$[0.496629, 0.506629]$	32,201	92,104

parameters. We have combined validated integration with δ -complete decision procedures for solving the probabilistic δ -reachability problem. Our technique computes a *numerically guaranteed* enclosure for the probabilities that the system reaches the unsafe region in a finite number of discrete transitions. For systems with continuous random (but no nondeterministic) parameters, such enclosure can be made arbitrarily small. We have implemented our technique in the open source tool **ProbReach** and have applied it to a number of case studies featuring highly nonlinear ODEs, unbounded continuous random parameters and nondeterministic parameters. We have validated our results against Monte Carlo simulation, and the comparison supports the correctness of our approach.

Our work shows that it is possible to verify bounded reachability for hybrid systems featuring continuous random and nondeterministic parameters with the same level of accuracy as for finite-state stochastic systems. Of course, more work needs to be done in terms of improving both the tool engineering and the theory. With respect to the former, a more efficient parallel strategy needs to be implemented, and more experiments need to be performed to assess better the tool scalability. For the theory, in the future we plan to tackle a larger class of hybrid systems, which in particular include state-dependent probabilistic jumps and continuous probabilistic dynamics (stochastic differential equations).

References

1. Abate, A., Katoen, J.P., Lygeros, J., Prandini, M.: Approximate model checking of stochastic hybrid systems. *European Journal of Control* 16(6), 624 – 641 (2010)
2. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.H.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: *Hybrid Systems*. LNCS, vol. 736, pp. 209–229 (1992)
3. Alur, R., Dill, D.L.: Automata for modeling real-time systems. In: *ICALP*. LNCS, vol. 443, pp. 322–335 (1990)
4. Brihaye, T., Doyen, L., Geeraerts, G., Ouaknine, J., Raskin, J., Worrell, J.: On reachability for hybrid automata over bounded time. In: *ICALP*. LNCS, vol. 6756, pp. 416–427 (2011)
5. David, A., Larsen, K., Legay, A., Mikučionis, M., Poulsen, D.B.: Uppaal SMC tutorial. *International Journal on Software Tools for Technology Transfer (STTT)* (2015), *to appear*.
6. Ellen, C., Gerwinn, S., Fränzle, M.: Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. *International Journal on Software Tools for Technology Transfer (STTT)* (2014), *to appear*.
7. Enszer, J.A., Stadtherr, M.A.: Verified solution and propagation of uncertainty in physiological models. *Reliable Computing* 15, 168–178 (2010)

8. Fränzle, M., Hahn, E.M., Hermanns, H., Wolovick, N., Zhang, L.: Measurability and safety verification for stochastic hybrid systems. In: HSCC. pp. 43–52 (2011)
9. Fränzle, M., Teige, T., Eggers, A.: Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *J. Log. Algebr. Program.* 79(7), 436–466 (2010)
10. Galdino, S.: Interval integration revisited. *Open Journal of Applied Sciences* 2(4B), 108–111 (2012)
11. Gao, S., Avigad, J., Clarke, E.M.: Delta-decidability over the reals. In: LICS. pp. 305–314 (2012)
12. Gao, S., Kong, S., Chen, W., Clarke, E.M.: Delta-complete analysis for bounded reachability of hybrid systems. *CoRR* arXiv:1404.7171 (2014)
13. Gao, S., Kong, S., Clarke, E.M.: dReal: An SMT solver for nonlinear theories over the reals. In: CADE-24. LNCS, vol. 7898, pp. 208–214 (2013)
14. Hoeffding, W.: Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* 58(301), 13–30 (1963)
15. Ideta, A.M., Tanaka, G., Takeuchi, T., Aihara, K.: A mathematical model of intermittent androgen suppression for prostate cancer. *Journal of Nonlinear Science* 18(6), 593–614 (2008)
16. Ko, K.I.: *Complexity Theory of Real Functions*. Birkhäuser (1991)
17. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV. LNCS, vol. 6806, pp. 585–591 (2011)
18. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)* 1, 134–152 (1997)
19. Lerch, M., Tischler, G., Gudenberg, J.W.V., Hofschuster, W.e., Krämer, W.: FILIB++, a fast interval library supporting containment computations. *ACM Trans. Math. Softw.* 32(2), 299–324 (2006)
20. Liu, B., Kong, S., Gao, S., Zuliani, P., Clarke, E.M.: Towards personalized cancer therapy using delta-reachability analysis. In: HSCC. pp. 227–232. ACM (2015)
21. Novak, E., Woźniakowski, H.: Relaxed verification for continuous problems. *Journal of Complexity* 8(2), 124 – 152 (1992)
22. Petras, K.: Principles of verified numerical integration. *Journal of Computational and Applied Mathematics* 199(2), 317 – 328 (2007)
23. Ramponi, F., Chatterjee, D., Summers, S., Lygeros, J.: On the connections between PCTL and dynamic programming. In: HSCC. pp. 253–262. ACM (2010)
24. Richardson, D.: Some undecidable problems involving elementary functions of a real variable. *J. Symb. Log.* 33(4), 514–520 (1968)
25. Shmarov, F., Zuliani, P.: ProbReach: Verified probabilistic δ -reachability for stochastic hybrid systems. In: HSCC. pp. 134–139. ACM (2015)
26. Song, B., Thomas, D.: Dynamics of starvation in humans. *Journal of Mathematical Biology* 54(1), 27–43 (2007)
27. Soudjani, S.E.Z., Gevaerts, C., Abate, A.: FAUST²: Formal abstractions of uncountable-state stochastic processes. In: TACAS (2015), *to appear*.
28. Tiwari, A.: Formal semantics and analysis methods for Simulink Stateflow models. Tech. rep., SRI International (2002), <http://www.csl.sri.com/users/tiwari/html/stateflow.html>
29. Tkachev, I., Abate, A.: Formula-free finite abstractions for linear temporal verification of stochastic hybrid systems. In: HSCC. pp. 283–292. ACM (2013)
30. Zhang, L., She, Z., Ratschan, S., Hermanns, H., Hahn, E.M.: Safety verification for probabilistic hybrid systems. In: CAV. LNCS, vol. 6174, pp. 196–211 (2010)

Appendix

A Proofs

Proof (Proposition 4). Immediate from the fact that (Definition 1) the sets defined by flow_q , init_q , jump_q , and unsafe_q are Borel, and conjunction and disjunctions correspond to set intersection and union, respectively. \square

Proof (Proposition 7). It was proven in [16, Corollary 6.3] that the complexity of computing derivatives of $f \in P_{C^5[a,b]}$ is P. Thus, computing a partial sum on an interval $[x]$ and evaluating the formula

$$\text{width}([K]([x])) > \epsilon \frac{\text{width}([x])}{b-a}$$

is also polynomial in time. Given an arbitrary partition containing n intervals, the formula above can thus be verified in polynomial time with respect to the size of the partition. Hence, obtaining a partition such that on each interval the formula above holds is in NP complexity class. \square

We recall that $P_{C^n[a,b]}$ denotes the class of polynomial-time (Type 2) computable functions whose derivative $f^{(n)}$ exists and is continuous over $[a, b]$. Ko [16, Section 6.2] showed that if f is also analytic, then integration becomes P. However, such an algorithm essentially uses truncated Taylor series over an arbitrary partition. Instead, our Algorithm 1 adaptively searches for a partition that guarantees the required error bound ϵ , while having a minimal number of intervals. In practice, this significantly benefits the performance of our whole implementation. Another advantage of Algorithm 1 is that it does not require f to be analytic.

Proof (Lemma 8). It was proven in [11] that satisfiability of a first-order formula implies satisfiability of its weakening. Therefore, following can be equivalently derived:

$$\phi \rightarrow \phi^{\delta'} \Leftrightarrow \neg\phi \vee \phi^{\delta'} \Leftrightarrow \phi^{\delta'} \vee \neg\phi \Leftrightarrow \neg\phi^{\delta'} \rightarrow \neg\phi$$

Let now $\psi = \phi^{\delta'}$ and $\psi^{\delta^*} = \phi^{\delta}$ be weakening of ψ . It was proven that if the weakening of the formula is unsatisfiable then the formula is also unsatisfiable. Then:

$$\neg\psi^{\delta^*} \rightarrow \neg\psi \Leftrightarrow \neg\phi^{\delta} \rightarrow \neg\phi^{\delta'}$$

\square

Proof (Proposition 9). By the definition of the decision procedure both formulas can be δ -sat on an interval if and only if the considered interval contains values from the Borel set B and its complement, or when a *false alarm* occurs. Then it can be concluded that the initial interval contains a subinterval which is either in the Borel set B or outside it. This can be stated as:

$$\begin{aligned} \exists \delta \in \mathbb{Q}^+ : (\phi([u, v]) - \delta\text{-sat}) \wedge (\phi^C([u, v]) - \delta\text{-sat}) \Rightarrow \\ \exists [u', v'] \subseteq [u, v] : ([u', v'] \cap B = [u', v']) \oplus ([u', v'] \cap B = \emptyset) \end{aligned} \quad (11)$$

Then applying the decision procedure to $[u', v']$ and decreasing δ , it is guaranteed that eventually we will obtain such a δ that the weakening of the formula will be false. In other words:

$$\exists \delta \in \mathbb{Q}^+ : (\neg(\phi([u', v']) - \delta\text{-sat})) \oplus (\neg(\phi^C([u', v']) - \delta\text{-sat})) \quad (12)$$

Therefore, by Lemma 8 the decision procedure will return **unsat** for one of the formulas in (12):

$$\begin{aligned} \exists \delta \in \mathbb{Q}^+ : (\neg(\phi([u', v']) - \delta\text{-sat})) \oplus (\neg(\phi^C([u', v']) - \delta\text{-sat})) \Rightarrow \\ (\phi([u', v']) - \text{unsat}) \oplus (\phi^C([u', v']) - \text{unsat}) \end{aligned} \quad (13)$$

□

Proof (Proposition 10).

If a hybrid system has l independent initial random parameters with bounded support, then the reachability probability can be computed as:

$$\int_B \prod_{i=1}^l dP_i(r_i) = \int_{\Omega} I_B(r_1, \dots, r_l) \prod_{i=1}^l dP_i(r_i) \quad (14)$$

where P_i is the probability measure of the i -th random parameter r_i , B is the Borel set (3) that contains all the random parameters values for which the hybrid system reaches the unsafe region in k steps, Ω is the domain of the random parameters, and $I_B(r_1, \dots, r_l)$ is the indicator function.

In order to compute (14) with precision ϵ_{prod} , we must be able to compute

$$\int_{\Omega} \prod_{i=1}^l dP_i(r_i) \quad (15)$$

with the same precision. By Fubini's theorem, integral (15) can be calculated as the product

$$\int_{\Omega} \prod_{i=1}^l dP_i(r_i) = \prod_{i=1}^l \int_{a_i}^{b_i} dP_i(r_i) = \prod_{i=1}^l I_i$$

where

$$I_i = \int_{a_i}^{b_i} dP_i(r_i)$$

and a_i, b_i are the domain bounds of random parameter r_i .

Now, we can compute an interval of length ϵ_i containing the *exact* value of each integral I_i , and let us denote such interval as $[\hat{I}_i, \hat{I}_i + \epsilon_i]$. It is thus sufficient to demonstrate how the values ϵ_i 's should be chosen in order for the integral (15) to be contained in an interval of length ϵ_{prod} .

According to the rules of interval arithmetics, product of the intervals is contained in the interval:

$$[\hat{I}_1, \hat{I}_1 + \epsilon_1] \cdot [\hat{I}_2, \hat{I}_2 + \epsilon_2] \cdots [\hat{I}_l, \hat{I}_l + \epsilon_l] \subseteq [\prod_{i=1}^l \hat{I}_i, \prod_{i=1}^l (\hat{I}_i + \epsilon_i)] \quad (16)$$

Therefore, the ϵ_i 's should be chosen such that the interval at the RHS of inclusion (16) has length smaller than ϵ_{prod} , i.e., the following should hold:

$$\prod_{i=1}^l (\hat{I}_i + \epsilon_i) - \prod_{i=1}^l \hat{I}_i \leq \epsilon_{prod} \quad (17)$$

Therefore, choosing ϵ_i in such a way that (17) holds will guarantee that the *exact* value of the product of l integrals is contained in the interval of size ϵ_{prod} . If we want all the ϵ_i 's equal to a single value ϵ , then formula (17) can be satisfied by assuming in the worst case $\hat{I}_i = 1$ for all i , which gives

$$\epsilon_{prod} \geq \prod_{i=1}^l (1 + \epsilon) - 1 = \sum_{i=1}^l \binom{l}{i} \epsilon^i$$

where $\binom{l}{i}$ is the binomial coefficient. □

Proposition 14. *Given $\epsilon \in (0, 1] \cap \mathbb{Q}$, $k \in \mathbb{N}$ and a hybrid system with one **bounded** continuous random initial parameter, there exists an algorithm for computing an interval of size not larger than ϵ that contains the value of (2), i.e., the probability of reaching the unsafe region in k steps.*

Proof. Let $r \in [a, b]$ be a random continuous parameter. Then by applying our validated integration procedure (Algorithm 1) we obtain a partition $\cup_{i=1}^n [r]_i$ such that on each of the intervals the value of the partial sum is enclosed by an interval of length $\epsilon \frac{\text{width}([r]_i)}{\text{width}([a, b])}$, and the value of the integral on $[a, b]$ is enclosed by the interval of size ϵ .

Let k -th step reachability be encoded by the formula ϕ , and ϕ^C be derived as in (9). By applying the decision procedure to all the intervals from the initial partition, we can distributed them in three sets B_{unsat} , $B_{C^{unsat}}$, $B_{\delta-sat}$ containing the intervals where ϕ is **unsat**, ϕ^C is **unsat**, and both formulas are **δ -sat**, respectively. Then the following will hold:

$$\int_a^b f(r) dr = \int_{B_{unsat}} f(r) dr + \int_{B_{C^{unsat}}} f(r) dr + \int_{B_{\delta-sat}} f(r) dr \quad (18)$$

The lower and the upper bounds of the interval containing the exact value of the probability can be found as:

$$P_{lower} = \int_{B_{C^{unsat}}} f(r) dr$$

$$P_{upper} = \int_a^b f(r) dr - \int_{B_{unsat}} f(r) dr$$

Then size of the interval $[P_{lower}, P_{upper}]$ can be calculated as:

$$\begin{aligned} P_{upper} - P_{lower} &= \int_a^b f(r) dr - \int_{B_{unsat}} f(r) dr - \int_{B_{Cunsat}} f(r) dr = \\ &= \int_{B_{\delta-sat}} f(r) dr \end{aligned}$$

By Proposition 9 it follows that on each interval $[u, v]$ in $B_{\delta-sat}$ we can obtain a subinterval $[u', v']$ such that it can be added to B_{unsat} or B_{Cunsat} and, thus, removed from $B_{\delta-sat}$. Therefore, as $\delta \rightarrow 0$ and $n \rightarrow \infty$ (where n is the number of disjoint subintervals partitioning $B_{\delta-sat}$) the size of set $B_{\delta-sat}$ will be decreasing. Hence, we can conclude that $\int_{B_{\delta-sat}} f(r) dr \rightarrow 0$, which implies:

$$\exists \epsilon \in \mathbb{Q}^+ : P_{upper} - P_{lower} \leq \epsilon$$

□

Proposition 15. *Given $\epsilon \in (0, 1] \cap \mathbb{Q}$, $k \in \mathbb{N}$ and a hybrid system with one **unbounded** continuous random initial parameter, there exists an algorithm for computing an interval of size not larger than ϵ that contains the value of (2).*

Proof. Let us recall that calculating the probability of reaching the unsafe region requires integrating an indicator function with respect the probability measure associated to the random parameter

$$\int_{\Omega} I_B(r) dP(r) \tag{19}$$

where $I_B(r)$ is the indicator function over set B (3), P is the probability measure of the random variable, and $\Omega = (-\infty, +\infty)$. In the following we shall simplify notation and write dP instead of $dP(r)$, since there is only one random variable.

The next inequality can be readily derived from the definition of indicator function:

$$0 \leq \int_{\Omega} I_B(r) dP \leq \int_{\Omega} dP.$$

By the property of definite integral, for any a and $b \geq a$:

$$\int_{\Omega} I_B(r) dP = \int_a^b I_B(r) dP + \int_{-\infty}^a I_B(r) dP + \int_b^{\infty} I_B(r) dP.$$

As $\int_{-\infty}^a I_B(r) dP \geq 0$ and $\int_b^{\infty} I_B(r) dP \geq 0$, the following holds for all $r \in \Omega$

$$\int_a^b I_B(r) dP \leq \int_{\Omega} I_B(r) dP \leq \int_a^b I_B(r) dP + 1 - \int_a^b dP.$$

Therefore, the exact value of probability is enclosed by the interval:

$$\int_{\Omega} I_B(r) dP \in [\int_a^b I_B(r) dP, \int_a^b I_B(r) dP + 1 - \int_a^b dP]$$

By Proposition 14 we can calculate the lower and the upper bounds of the probability over the bounded interval $[a, b]$:

$$\int_{\Omega} I_B(r) dP \in [(\int_a^b I_B(r) dP)_{lower}, (\int_a^b I_B(r) dP)_{upper} + 1 - \int_a^b dP] \quad (20)$$

Now it is desired that the interval in formula (20) is of length ϵ . For this the error ϵ can be presented as a sum of two components ϵ_{inf} and ϵ_{prob} that are chosen such that: $\epsilon \geq \epsilon_{inf} + \epsilon_{prob}$ where $\epsilon_{inf} \geq 1 - \int_a^b dP$ and $\epsilon_{prob} \geq (\int_a^b I_B(r) dP)_{upper} - (\int_a^b I_B(r) dP)_{lower}$.

The values a and b can be obtained by solving the first inequality as a first order formula:

$$\exists a \in [u_a, v_a], \exists b \in [u_b, v_b] : (\frac{dF}{dx} = f(x)) \wedge (F(a) = 0) \wedge (F(b) \geq 1 - \epsilon_{inf}) \quad (21)$$

where f is the probability density function of the random parameter, which is known to the user. (Note that F thus denotes the cumulative distribution function of the random parameter.) Then the values a and b derived from formula (21) are used to compute the interval $[(\int_a^b I_B(r) dP)_{lower}, (\int_a^b I_B(r) dP)_{upper}]$ of length ϵ_{prob} . This can be performed for an arbitrary positive rational number (by Proposition 14).

If formula (21) is unsatisfiable then it means that bounds for the variables a and b should be enlarged and the formula should be verified again. This process should repeat until the formula is satisfiable and the values a and b are obtained. \square

Proof (Proposition 11). Evaluating formulas ϕ and ϕ^C on two boxes $[\bar{r}]$ and $[\bar{z}]$ (over random and nondeterministic continuous parameters, respectively) there are four possible outcomes:

- $\phi([\bar{r}], [\bar{z}])$ is **unsat**. Hence, there are *for sure* no values in $[\bar{r}]$ and $[\bar{z}]$ such that the system reaches the unsafe region, so $[\bar{r}]$ is not in B .
- $\phi([\bar{r}], [\bar{z}])$ is **δ -sat**. Then, there is a value in $[\bar{r}], [\bar{z}]$ such that the system reaches U or U^δ (δ -weakening of set U).
- $\phi^C([\bar{r}], [\bar{z}])$ is **unsat**. Therefore, there is *for sure* no value in $[\bar{r}]$ and $[\bar{z}]$ such that for all time points on the k -th step the system stays within the complement of the unsafe region. In other words, for all the values in $[\mathbf{r}]$ the system reaches the unsafe region, so $[\mathbf{r}]$ is fully contained in B .
- $\phi^C([\bar{r}], [\bar{z}])$ is **δ -sat**. Then there is a value in $[\bar{r}], [\bar{z}]$ such that the system stays within U^C or U^{C^δ} .

Similarly to the approach used in the proof of Proposition 14, lower and upper bounds of the reachability probability can be calculated as:

$$P_{lower} = \int_{B_{C_{unsat}}} f(r) dr$$

$$P_{upper} = \int_a^b f(r) dr - \int_{B_{unsat}} f(r) dr$$

where $B_{unsat}, B_{C^{unsat}}, B_{\delta-sat}$ containing the boxes where ϕ is **unsat**, ϕ^C is **unsat**, and both formulas are δ -**sat** respectively. Hence, by refining boxes from $B_{\delta-sat}$ until $\max(|[r]|) \leq \epsilon$, we obtain an interval $[P_{lower}, P_{upper}]$ containing the range of probabilities of reaching the unsafe region. \square

Proof (Theorem 12). Let ϕ be a formula describing a hybrid system with discrete random parameters $\{D_1, \dots, D_p\}$, and let $p(\cdot)$ denotes their probabilities, *i.e.*, for each D_i we have that $\sum_{j=1}^{\#D_i} p(d_{ij}) = 1$.

Let $\mathbf{DD} = D_1 \times \dots \times D_p$ be the Cartesian product of discrete parameters. For each $\mathbf{dd} = \{d_{11}, d_{22}, \dots, d_{pk}\} \in \mathbf{DD}$, let $m_{\mathbf{dd}} = p(d_{11}) \cdot p(d_{21}) \cdot \dots \cdot p(d_{pk})$. Substituting all discrete random parameters with their values from \mathbf{dd} we will obtain a hybrid system which can be described by a corresponding formula $\phi_{\mathbf{dd}}$.

Now depending of the type of the considered hybrid system we can use one of the algorithms already presented.

- **PHA** or **NPHA**: we can apply Algorithm 2 and obtain a probability interval $[[P_{lower}], [P_{upper}]]_{\mathbf{dd}}$.
- **HA**: we can just use the decision procedure described above and evaluate $\phi_{\mathbf{dd}}$ and $\phi_{\mathbf{dd}}^C$. Then returned value depends on the evaluation outcome:
 - $\phi_{\mathbf{dd}}$ -**unsat** return $[[P_{lower}], [P_{upper}]]_{\mathbf{dd}} = [0.0, 0.0]$
 - $\phi_{\mathbf{dd}}^C$ -**unsat** return $[[P_{lower}], [P_{upper}]]_{\mathbf{dd}} = [1.0, 1.0]$
 - $\phi_{\mathbf{dd}}$ - δ -**sat** and $\phi_{\mathbf{dd}}^C$ - δ -**sat** return $[[P_{lower}], [P_{upper}]]_{\mathbf{dd}} = [0.0, 1.0]$

Doing so for each $\mathbf{dd} \in \mathbf{DD}$ we can obtain the resulting probability interval

$$[P] = \sum_{\mathbf{dd} \in \mathbf{DD}} (m_{\mathbf{dd}} \cdot [[P_{lower}], [P_{upper}]]_{\mathbf{dd}})$$

\square

Proof (Theorem 13). The considered algorithm can be presented as two independent components: validated integration and probability calculation.

The decision procedure used in the algorithm consists of two formulas: ϕ and ϕ^C , which are Σ_1 and Σ_2 sentences. Solving these formulas as a δ -SMT problem is in $(\Sigma_1^P)^C$ and $(\Sigma_2^P)^C$ complexity classes respectively, where $P \subseteq C \subseteq PSPACE$ is the complexity of the terms in the formula [11]. Hence, the decision procedure is in $(\Sigma_2^P)^C$. Then verification of an arbitrary partition of n intervals is also in $(\Sigma_2^P)^C$. Hence, it is clear that obtaining the *correct* partition (such that $[P_{upper}] - [P_{lower}] \leq \epsilon_{prob}$) is in $NP^{(\Sigma_2^P)^C}$. By Proposition 7 the complexity of the verified integration is NP , which is in $NP^{(\Sigma_2^P)^C}$. The complexity of the whole algorithm is thus $NP^{(\Sigma_2^P)^C}$, where $P \subseteq C \subseteq PSPACE$.

Finally, it has been shown in [11] that if ϕ (and thus ϕ^C) includes Lipschitz-continuous ODEs then the δ -SMT problem becomes $PSPACE$ -complete. This lifts the complexity of the whole algorithm to $PSPACE$ -complete. \square

B Models

We give here more information about the models used for our experiments.

B.1 2D-moving Bouncing ball

The ball is launched from position $(S_x \in [-5, 5], S_y = 0)$ with initial speed $v_0 \sim N(20, 1)$, *i.e.*, normal distribution with mean 20 and variance 1, and angle α to horizon (measured in radians) with the following probability distribution:

$$\begin{aligned} P[\alpha = 0.5236] &= 0.3 \\ P[\alpha = 0.7854] &= 0.5 \\ P[\alpha = 1.0472] &= 0.2 \end{aligned}$$

After each jump the speed of the ball is multiplied by 0.9. The gravity of Earth parameter $g \in [9.8, 9.81]$ is also nondeterministic. The system is modelled as a hybrid system with one mode with dynamics governed by a system of ODEs:

$$\begin{aligned} S'_x(t) &= v_0 \cos \alpha \\ S'_y(t) &= v_0 \sin \alpha - gt \end{aligned}$$

The goal of the experiment is to calculate the probability of reaching the region $S_x(t) \geq 100$ within 0 and 1 jump. The results are presented in Table 4. Monte Carlo simulation of continuous nondeterminism in MATLAB was achieved as explained in Section 5, using uniform discretisation of the domains of the non-deterministic parameters (S_x and g were discretised with 100 and 10 values, respectively). In Figure 1 and 2 we plot the Monte Carlo reachability probability estimate with respect to the nondeterministic parameters $S_x(0)$ and g , for 0 and 1 jump, respectively.

Table 4. 2D-moving bouncing ball model; see legend in Section 5.

Method	Model type	k	ϵ	$length$	Probability interval	CPU
Prob	NPHA	0	10^{-3}	$2.38 \cdot 10^{-4}$	[0.000013103, 0.000250681]	223
Reach	NPHA	1	10^{-3}	$6.464 \cdot 10^{-2}$	[0.0647381, 0.12937951]	1,605

Method	Model type	k	ζ	c	Monte Carlo interval	CPU	N
Monte	NPHA	0	$5 \cdot 10^{-3}$	0.99	[0, 0.00520629]	1,482	92,104
Carlo	NPHA	1	$5 \cdot 10^{-3}$	0.99	[0.0585, 0.1367]	1,485	92,104

B.2 Starvation model

In humans, enduring fasting for 3-4 days will consume all the glucose reserves of the body. At this point, the energy to sustain the human body is produced

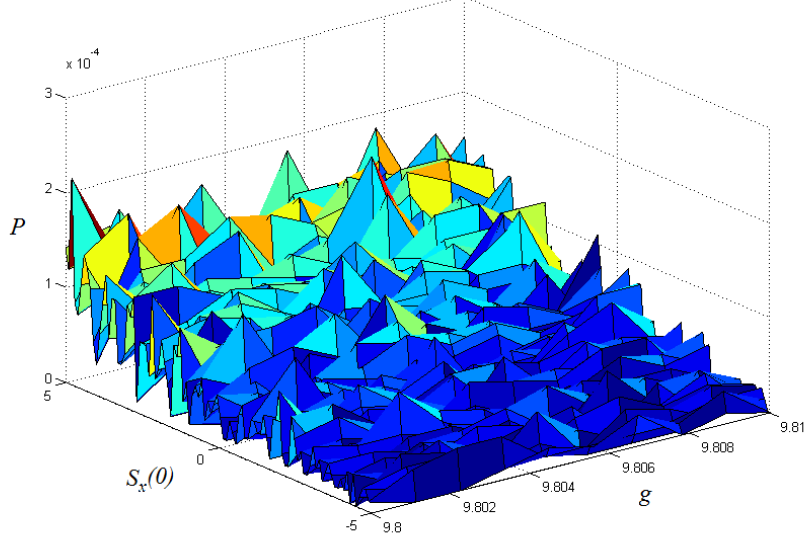


Fig. 1. Monte Carlo simulation of the bouncing ball model: Reachability probability ($k = 0$) estimate P with respect to nondeterministic parameters $S_x(0) \in [-5, 5]$ and $g \in [9.8, 9.81]$.

from fat $F(t)$, muscles $M(t)$ and ketone bodies $K(t)$ (for brain function) [26]. The ODE system below represents the dynamics of the described variables:

$$\begin{aligned}\frac{dF}{dt} &= F\left(\frac{-a}{1+K} - \frac{1}{\lambda_F}\left(\frac{C+gL_0}{F+M} + g\right)\right) \\ \frac{dM}{dt} &= -\frac{M}{\lambda_M}\left(\frac{C+gL_0}{F+M} + g\right) \\ \frac{dK}{dt} &= \frac{VaF}{1+K} - b\end{aligned}$$

We consider two scenarios where parameter $g \sim N(10.96, 1)$, *i.e.*, normally distributed with mean 10.96 and variance 1, and:

- $b \in [0.05, 0.075]$ is nondeterministic; or
- b is a discrete random parameter with the probability distribution:

$$\begin{aligned}P[b = 0.05] &= 0.1 \\ P[b = 0.06] &= 0.2 \\ P[b = 0.07] &= 0.3 \\ P[b = 0.075] &= 0.4\end{aligned}$$

The probabilistic reachability property investigated in the experiment is: *what is the probability that muscle mass will decrease by 40% within 25 days?*

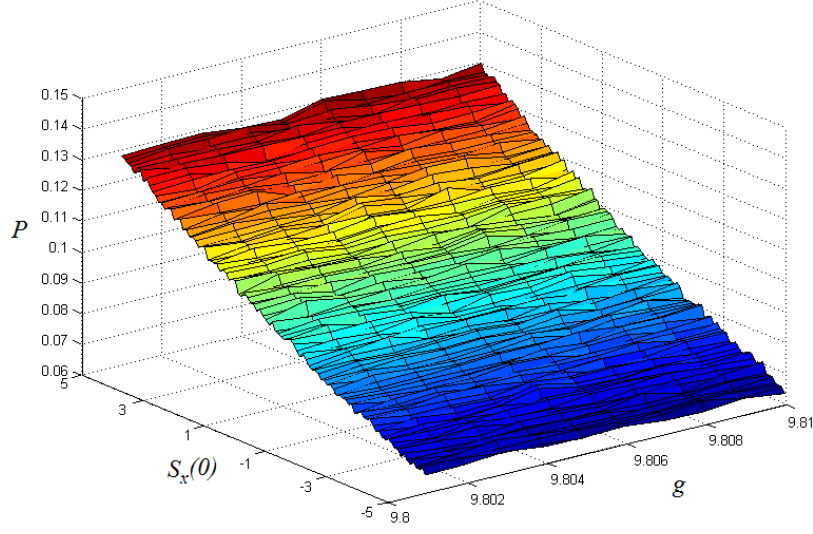


Fig. 2. Monte Carlo simulation of the bouncing ball model: Reachability probability ($k = 1$) estimate P with respect to nondeterministic parameters $S_x(0) \in [-5, 5]$ and $g \in [9.8, 9.81]$.

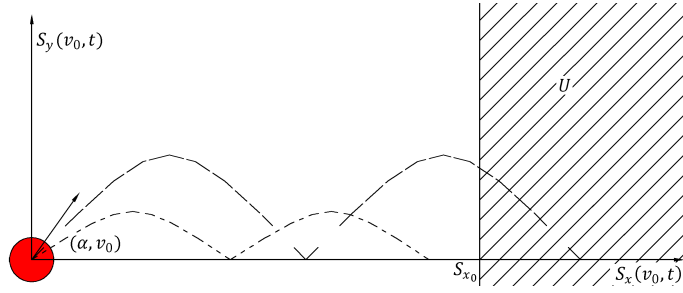


Fig. 3. 2D-moving bouncing ball scenario

Numerical values for all deterministic parameters in the model are presented in Table 5 and verification results are featured in Table 1. Monte Carlo simulation of continuous nondeterminism in MATLAB was achieved as explained in Section 5, via uniform discretisation (10 values) of the nondeterministic parameter b . In Figure 4 we plot the Monte Carlo reachability probability and confidence interval with respect to the value of parameter b .

Table 5. Starvation model parameters and initial conditions

Param.	Value	Param.	Value	Param.	Value	Param.	Value	Param.	Value
a	0.013	λ_F	7777.8	$M(0)$	43.6	V	0.9	$K(0)$	0.02
C	772.3	λ_M	1400	$F(0)$	25	L_0	30.4		

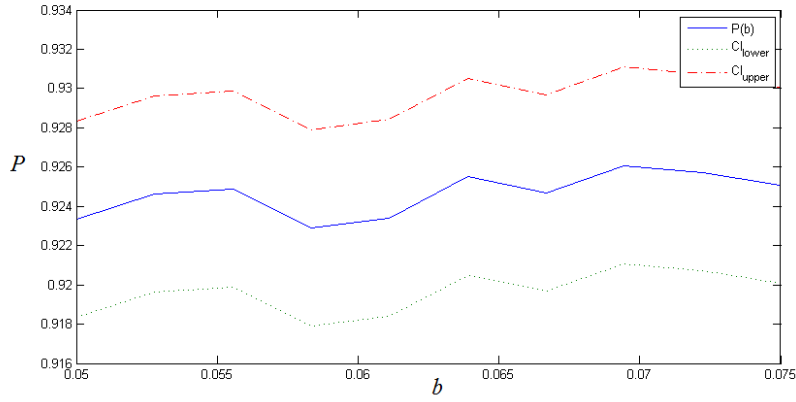


Fig. 4. Monte Carlo simulation of the starvation model: Reachability probability estimate P (solid line) with respect to nondeterministic parameter $b \in [0.05, 0.075]$. For each (discretised) value of b we give a Chernoff-Hoeffding confidence interval, denoted by dotted lines.

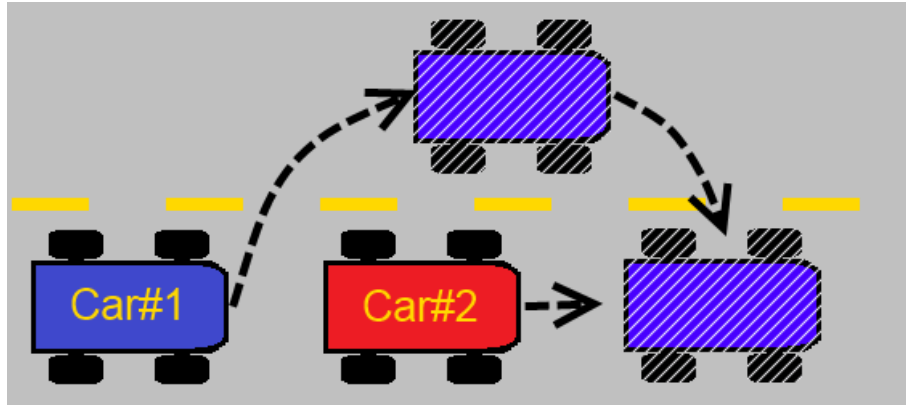


Fig. 5. Road scenario

B.3 Road scenario

We consider the road scenario inspired by a model presented in [6] and depicted in Figure 5. Two cars (*Car#1* and *Car#2*) move on the same lane, starting at coordinates $S_{01} = 0$ and $S_{02} = S_{01} + v_1 \cdot t_{safe}$, where $t_{safe} = 2sec$ implements the so-called “two seconds rule” for maintaining a safety distance between two cars.

We describe a car collision scenario and we model it with the hybrid automaton given in Figure 6. Starting in Mode 1 at time $t = 0$, *Car#1* changes lane and starts accelerating at $a_{a1} \text{ m/s}^2$, while *Car#2* is moving in the initial lane with speed v_2 . Upon reaching the maximum speed v_{max} , the system switches to Mode 2, where *Car#1* keeps moving at this speed until it gets ahead of *Car#2* by the safety distance $S_{safe} = v_2 \cdot t_{safe}$. After that, we switch to Mode 3: *Car#1* returns to the initial lane and starts decelerating at a_{d1} . For the driver of *Car#2* it takes $t_{react} = 1sec$ to react (the system switches to Mode 4) and then it starts decelerating as well (with random acceleration $a_{d2} \sim N(-1.35, 0.01)$). In Mode 3, 4, and 5 we also have an invariant specifying that *Car#1* should precede *Car#2* at all time. We calculate the *probability of observing a car collision in Mode 5*, where *Car#1* is stopped. Numerical values for all deterministic parameters in the model are given in Table 6 and verification results are presented in Table 3.

Table 6. Car collision model parameters and initial conditions

Param.	Value	Param.	Value	Param.	Value	Param.	Value	Param.	Value
v_1	11.12	v_2	11.12	v_{max}	16.67	t_{safe}	2	S_{safe}	$v_2 \cdot t_{safe}$
a_{a1}	3	a_{d1}	-4	t_{react}	1	S_{01}	0	S_{02}	$S_{01} + v_1 \cdot t_{safe}$

B.4 Prostate cancer therapy

We consider a model of personalised prostate cancer therapy introduced by Ideta *et al.* [15] and improved by Liu *et al.* [20]. Intermittent androgen suppression (IAS) has proved to be more effective than constant androgen suppression (CAS) in delaying the recurrence of prostate cancer. Briefly, the personalised therapy comprises of two repeating stages. The patient’s prostate-specific antigen (PSA) level is monitored throughout the therapy. When the PSA level reaches an upper threshold, the patient starts receiving treatment (*on-therapy* stage) until the PSA level decreases to a lower threshold (*off-therapy*). The main aim of the therapy is to delay cancer relapse for as long as possible.

The model of the therapy is given in Figure 8 (a full explanation of the model and its parameters can be found in [20]). Mode 1 is the *on-therapy* stage, and it continues until the PSA level (measured by $x + y$) is above threshold $r_0 = 4$.

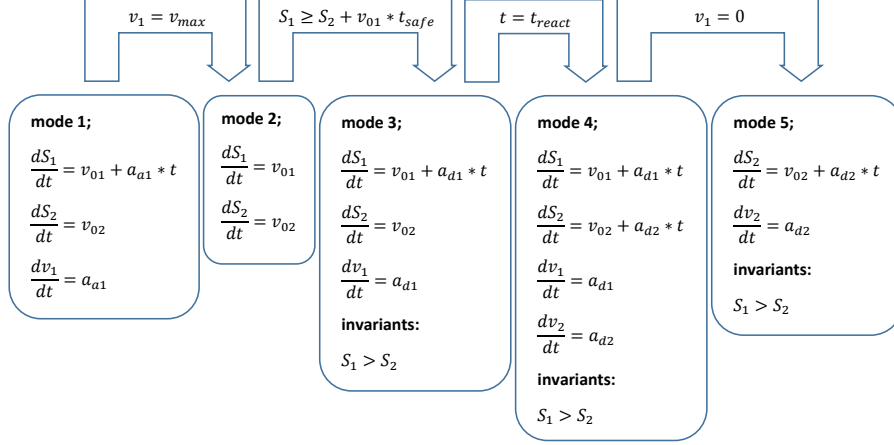


Fig. 6. Car collision model

Then the system makes a transition to the *off-therapy* mode which continues until PSA level is below $r_1 = 10$. We explore the following scenarios:

- α_y is distributed normally ($N(0.05, 0.01)$) and $\alpha_x = 0.0197$
- α_y is distributed normally ($N(0.05, 0.01)$) and $\alpha_x \in [0.0197, 0.0204]$ is non-deterministic

For the cases above we calculate the probability of cancer relapse (*i.e.*, $y \geq 1$) within 100 days of using the personalised cancer therapy. Numerical values of all the parameters in the model are presented in the Table 7 and verification results are featured in Table 2. Monte Carlo simulation of continuous nondeterminism in MATLAB was achieved as detailed in Section 5, using uniform discretisation (20 values) of the domain of the nondeterministic parameter α_x . In Figure 7 we plot Monte Carlo reachability probability and confidence interval with respect to the value of parameter α_x .

Table 7. Prostate cancer therapy model parameters and initial conditions

Param.	Value	Param.	Value	Param.	Value	Param.	Value	Param.	Value
β_x	0.0175	β_y	0.0168	k_1	10.0	k_2	1.0	k_3	10.0
k_4	2	m_1	10^{-5}	z_0	12	γ	0.08	r_1	10.0
r_0	4.0	d_0	1.0	c_1	0.01	c_2	0.03	c_3	0.02
$x(0)$	19	$y(0)$	0.1	$z(0)$	12.5				

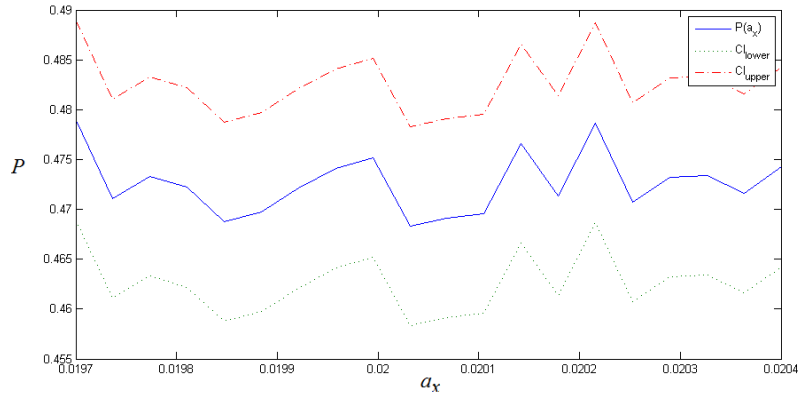


Fig. 7. Monte Carlo simulation of the prostate cancer therapy model: Reachability probability estimate P (solid line) with respect to the nondeterministic parameter $\alpha_x \in [0.0197, 0.0204]$. For each (discretised) value of α_x we give a Chernoff-Hoeffding confidence interval, denoted by dotted lines.

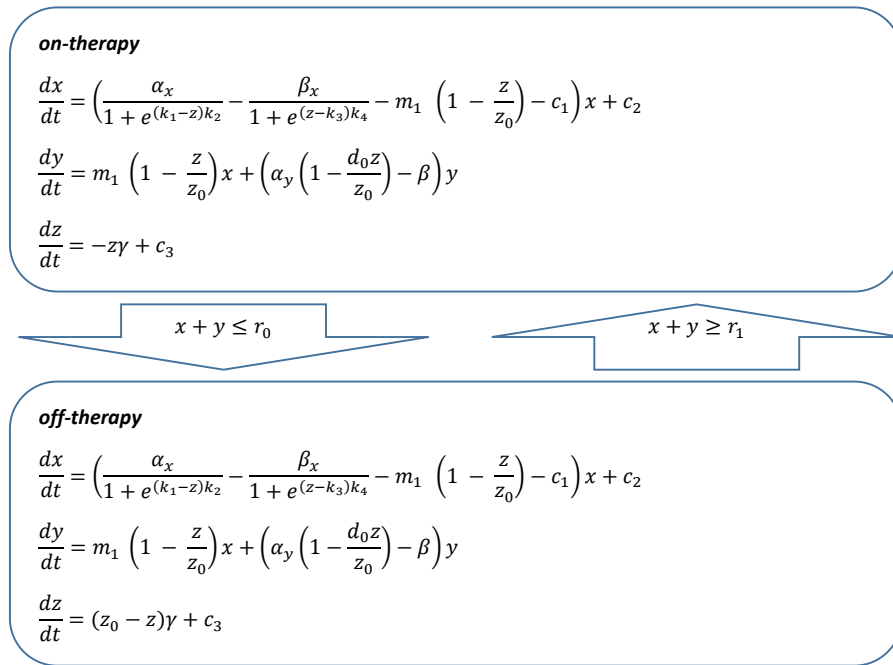


Fig. 8. Personalized prostate cancer therapy model

B.5 Prostate cancer therapy: ProbReach file

```

1 // This is a pdrh file corresponding to the prostate cancer therapy model with
2 // one random and one nondeterministic parameter.
3 MODEL_TYPE(NPHA) // defining model type
4 #define betax 0.0175
5 #define betay 0.0168
6 #define k1 10.0
7 #define k2 1.0
8 #define k3 10.0
9 #define k4 2
10 #define m1 0.00005
11 #define z0 12.0
12 #define gamma 0.08
13 #define r1 10.0
14 #define r0 4.0
15 #define d0 1.0
16 #define c1 0.01
17 #define c2 0.03
18 #define c3 0.02
19 #define Gx ((alphax/(1+exp((k1-z)*k2)))-(betax/(1+exp((z-k3)*k4))))
20 #define Gy ((alphay * (1 - (d0 * (z / z0)))) - betay)
21 #define Mxy (m1 * (1 - (z / z0)))
22 #define scale 1.0
23 #define T 100.0
24 N(0.05,0.01)alphax; // random parameter, normally distributed
25 [0,T]time;
26 [0,T]tau;
27 [0,100.0]x;
28 [0,10.0]y;
29 [0.0,100.0]z;
30 [0.0197,0.0204]alphax; // nondeterministic parameter
31 {
32 mode1; // on-therapy
33   invt:
34     (y <= 1);
35   flow:
36     d/dt[x]=scale * ((Gx - Mxy - c1) * x + c2);
37     d/dt[y]=scale * (Mxy * x + Gy * y);
38     d/dt[z]=scale * (-z * gamma + c3);
39     d/dt[tau]=scale * 1.0;
40   jump:
41     ((x+y)=r0)==>@2(and(tau'=tau)(x'=x)(y'=y)(z'=z));
42 }
43 {
44 mode2; // off-therapy
45   invt:
46     (y <= 1);
47   flow:
48     d/dt[x]=scale * ((Gx - Mxy - c1) * x + c2);
49     d/dt[y]=scale * (Mxy * x + Gy * y);
50     d/dt[z]=scale * ((z0 - z) * gamma + c3);
51     d/dt[tau]=scale * 1.0;
52   jump:
53     ((x+y)=r1)==>@1(and(tau'=tau)(x'=x)(y'=y)(z'=z));
54 }
55 init:
56   @1(and (x = 19) (y = 0.1) (z = 12.5) (tau = 0));
57 goal: // unsafe region
58   @2(and(y <= 1)(tau = T));
59 goal_c: // unsafe region complement
60   @2(and(y > 1.0)(tau < T));

```

C δ -satisfiability

In order to overcome the undecidability of reasoning about general real formulae, Gao *et al.* recently defined the concept of δ -satisfiability over the reals [11], and presented a corresponding δ -complete decision procedure. The main idea is to decide correctly whether slightly *relaxed* sentences over the reals are satisfiable or not. The following definitions are from [11].

Definition 16. A bounded quantifier is one of the following:

$$\begin{aligned}\exists^{[a,b]}x &= \exists x : (a \leq x \wedge x \leq b) \\ \forall^{[a,b]}x &= \forall x : (a \leq x \wedge x \leq b)\end{aligned}$$

Definition 17. A bounded Σ_1 sentence is an expression of the form:

$$\exists^{I_1}x_1, \dots, \exists^{I_n}x_n : \psi(x_1, \dots, x_n)$$

where $I_i = [a_i, b_i]$ are intervals, $\psi(x_1, \dots, x_n)$ is a Boolean combination of atomic formulas of the form $g(x_1, \dots, x_n) \text{ op } 0$, where g is a composition of Type 2-computable functions and $\text{op} \in \{<, \leq, >, \geq, =, \neq\}$.

Any bounded Σ_1 sentence is equivalent to a Σ_1 sentence in which all the atoms are of the form $f(x_1, \dots, x_n) = 0$ (i.e., the only op needed is ‘=’) [11]. Essentially, Type 2-computable functions can be approximated arbitrarily well by finite computations of a special kind of Turing machines (Type 2 machines); most of the ‘useful’ functions over the reals (e.g., continuous functions) are Type 2-computable [16].

The notion of δ -weakening [11] of a bounded sentence is central to δ -satisfiability.

Definition 18. Let $\delta \in \mathbb{Q}^+ \cup \{0\}$ be a constant and ϕ a bounded Σ_1 -sentence in the standard form

$$\phi = \exists^{I_1}x_1, \dots, \exists^{I_n}x_n : \bigwedge_{i=1}^m \left(\bigvee_{j=1}^{k_i} f_{ij}(x_1, \dots, x_n) = 0 \right) \quad (22)$$

where $f_{ij}(x_1, \dots, x_n) = 0$ are atomic formulas. The δ -weakening of ϕ is the formula:

$$\phi^\delta = \exists^{I_1}x_1, \dots, \exists^{I_n}x_n : \bigwedge_{i=1}^m \left(\bigvee_{j=1}^{k_i} |f_{ij}(x_1, \dots, x_n)| \leq \delta \right)$$

Note that ϕ implies ϕ^δ , while the converse is obviously not true. The bounded δ -satisfiability problem asks for the following: given a sentence of the form (22) and $\delta \in \mathbb{Q}^+$, correctly decide between

- **unsat:** ϕ is false,
- **δ -true:** ϕ^δ is true.

If the two cases overlap (i.e., ϕ is both false and δ -satisfiable) then either decision can be returned, thereby causing a *false alarm*. Such a scenario reveals that the formula is *fragile* — a small perturbation (i.e., a small δ) can change the formula’s truth value. The dReal tool [13] implements an algorithm for solving the δ -satisfiability problem, i.e., a δ -complete decision procedure. Basically, the algorithm combines a DPLL procedure (for handling the Boolean parts of the formula) with interval constraint propagation (for handling the real arithmetic atoms).